ORDERING IN A SEQUENTIAL STORAGE SYSTEM

Michael Lynn Mulhern

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93940

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ORDERING IN A SEQUENTIAL STORAGE SYSTEM

by

Michael Lynn Mulhern

Thesis Advisor: J.G. Taylor

March 1974

Approved for public release; distribution unlimited.

T 159107

Ordering in a Sequential Storage System

by

Michael Lynn Mulhern
Captain, United States Army
B.S., The Creighton University, 1967

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1974

Thesis
M8856
c.2

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93940

ABSTRACT

A missile system in which L launchers were available in a row and S spare missiles were stored in a continuation of this row was considered. Additionally, an assortment of warheads was available, but only one could be mounted in each missile. The ability of this system to meet a demand for a particular warhead was dependent on the mix of warheads in the L+S missiles and the order in which the missiles with their varying warheads were mounted on the launchers and in storage. Since the missiles were in a continuous row, some missiles would not be available for firing before the missiles in front of them had been fired.

Because of the myriad possible permutations of the L+S warheads, determination of optimal ordering of the warheads by simulation was found to be too time-consuming. Hence, analytic expressions were developed after an appropriate measure of effectiveness was devised. A computer program was developed to perform the recursive calculations necessary.

TABLE OF CONTENTS

LIST OF TABLES

SYMBOLS

a_i, a_{ji}, a_{kji}	number of targets missed due to availability between the i^{th} and $(i-1)^{st}$ reload, in the j^{th} section, in the k^{th} battery
a^f, a_j^f, a_{kj}^f	number of targets missed after the storage barn was emptied, by the j^{th} section, in the k^{th} battery
A_S, A_B	number of targets missed due to availability in a section and a battery respectively.
B	number of batteries in a defense
B_S, B_B	number of targets missed due to reloading in a section and a battery respectively
$F_o(x)$	completely specified cumulative distribution function
L	number of launchers in a section
m_i, m_{ji}, m_{kji}	number of targets missed during the i^{th} reload, in the j^{th} section, in the k^{th} battery
M_S, M_B, M_D	total number of targets missed by a section, a battery, and a defense respectively
M_{max}	maximum number of targets missed
M_{min}	minimum number of targets missed
p_j	probability of a request for a type j warhead
$q_k, q_{k\ell}$	weighting factor for a target missed due to availability by battery k, when it is of type ℓ
S	number of missiles in storage in a section
$s_k, s_{k\ell}$	weighting factor for a target missed due to reloading by battery k, when it is of type ℓ
$S_{15,000}(x)$	observed cumulative distribution function of a random sample of 15,000 observations

T	number of reloads required in a section
x_i	type of warhead stored on launcher i, $i = 1,2,\ldots,9$. $x_i \in \{1,2,3,4,5\}$
y_1, y_2, y_3	probability that a request is for the warhead on launcher #1, #2, or #3 respectively. $y_i = p_j$ if $x_i = j$, $i = 1,2,3$

I. INTRODUCTION

Current articles on missile allocation problems make certain tacit assumptions. The primary assumption is that all missiles are mounted on their own launchers. Thus, the problem of determining an optimal mix of missiles becomes one of determining the optimal mix of all missiles emplaced. However, a question comes to mind: What if emplacing each missile on its own launcher is infeasible for reasons of cost, mobility, maintenance, etc.? Then the missile complex will consist of a certain number of missiles mounted on launchers with the remainder in storage. The problem then becomes two problems: Should the mix of missiles on the launchers be the same as the mix of missiles in storage? If not, what should the respective mixes be? The two-part problem could be simplified back to the one-part problem by assuming that the time necessary to move the missiles in storage to launching positions will be so long as to render the stored missiles unavailable to any short run situation, that is, the storage is a reserve for future battles. However, if the system is set up so that the stored missiles can indeed be made available relatively quickly, the problem of respective mixes becomes crucial.

A further complication also arises when the stored missiles are not immediately available for reloading, but are placed in a sequential storage system. An analogy would be a magazine-fed rifle with a variety of types of rounds

available. The problem would consist not only of the mix of rounds in the magazine, but also the order in which the rounds are stored in the magazine. The order is critical because the rounds are fired sequentially. The situation can be understood by the following example: Suppose the magazine holds ten rounds and the only round of a certain type is placed tenth in the magazine. When the firer wants to fire that round, he must either fire or eject all of the intervening rounds. In either case, poor utilization of resources was caused by the ordering of rounds in the magazine.

Returning to the missile problem, it is not likely that a variety of types of missiles would be stored or emplaced together. However, the same situation would arise if one type of missile could be fitted with one of a variety of warheads. Even though the missiles would have identical external configurations, they could be considered different missiles because their use would vary with the warhead carried.

In summary, a missile site in which the missiles could be equipped with a variety of warheads and in which the missiles are placed in a ready storage area with a sequential reloading mechanism appears to present difficulties in allocation that are not present in systems commonly studied in the current literature.

Consequently, this thesis will attempt to establish a methodology for evaluation of missile sites designed as

described above. The first step will be to establish a measure of effectiveness suitable for such a system. When this has been established, further questions can be answered. The first question considered will be: Does ordering of warheads have any significant effect? If not, the question then becomes an academic exercise. If ordering is indeed significant (as it intuitively appears to be), further characteristics such as optimal ordering, optimal mix, or suboptimization, can be explored.

Since a general system will be considered, no definitive, real-world answers will be obtained. However, by exploring different possibilities, directions for further research will be indicated. In short, this thesis will describe the necessity of a separate methodology for evaluation of sequential ordered systems.

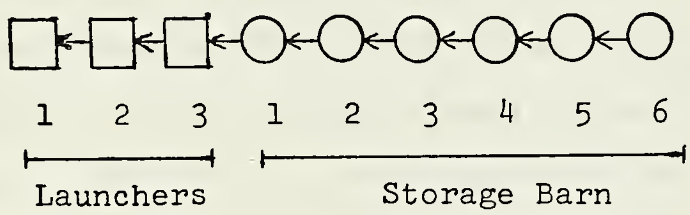
II. THE PHYSICAL SYSTEM

For this study, the launching area consists of three separate sections of three launchers and a storage barn holding six missiles. Thus the entire launching area (one battery) contains twenty-seven missiles when fully loaded. Further, it is assumed that five separate types of warheads can be installed in each missile [Headquarters, Department of the Army 1968]. Additionally, the mix of warheads in this battery is considered to be determined by some other agency. Any warhead can be mounted on any missile with all missiles completely identical in external configuration and hence, no specific considerations need be given about placing a certain type of warhead in any particular one of the twenty-seven available missiles. However, once emplaced, each missile has utility dependent upon which type of warhead is installed and where the missile is located in its section.

An additional restriction upon availability of missiles is that a missile in a section can be moved to another section only at a time expense of several hours since the missile has to be manually disassembled, moved to its new section, and reassembled. For safety reasons then, a section can not fire while a missile is being stocked. Another feature is that men must manually move the missiles from the storage locations onto the launchers during the reloading process. Hence, a section is out of action while either

reloading or restocking is accomplished. The launching area has enough personnel to conduct reloading operations in all three sections simultaneously. However, only one missile can be moved at a time in each section. The three sections are separated far enough that two sections can be reloading when the third section is firing. Since the restocking operation is so time consuming, its ramifications are considered to be strategic, rather than tactical, and are not considered in this analysis.

Because of the sequential organization, each missile has to be moved across all positions between their storage position and the destined launcher. This is best illustrated by an example:



Suppose the missile in storage #5 has to be moved to launcher #2. The missile has to move across storage positions #4, #3, #2 and #1 as well as launcher #3. For this to be feasible, the five intervening positions have to be empty. The only way this can occur is if the missiles had been fired, i.e. no means exists of disposing of a missile preventing a desired move. These missiles could have been fired only from the launchers. Hence, the four empty positions in the storage area indicated that previous to this

move, the first four missiles had been moved to the launchers and fired.

Consequently, if a desired warhead is in storage #4 and storages #3, #2, and #1 are full, the desired warhead would be inaccessible and, in effect, nonexistent for the engagement. This inaccessibility is obvious when it is considered that at best (for this situation) all three launchers could be empty. If this was the case, the missiles in storages #3, #2, and #1 could be moved onto the launchers. But then the desired missile in storage #4 still could not be fired because all launchers would be full. If one of these missiles was fired, the desired missile could then be moved onto a launcher. In summary, one missile has to be fired before the desired missile could possibly be fired.

The decision as to which of the five types of warheads available is desired for each firing is made external to the launching area. Due to this, the decision in the launching area consists of either being able to satisfy the request for a warhead or not. This capability is dependent upon the total mix of warheads in the launching area, and the order in which the warheads (in their missiles) are placed in the three launching sections. It is also assumed that each request for a warhead is independent of all others and that the p_j are available from the intelligence community.

The over-all firing pattern for this system is a shoot-look-shoot technique. That is, after a missile is fired,

the results of its detonation are assessed before the next round is fired. Thus the battery would have only one missile in the air at a given time. This has the effect that the battery engages only one target at a time no matter how many are within range. Thus, if many mobile targets are available, the battery would engage them serially until: (1) all targets are destroyed, (2) no targets are left within range, (3) no missiles are available on the launchers (reloading would enable reengagement), or (4) all missiles have been fired.

III. MEASURE OF EFFECTIVENESS

A measure of effectiveness must concern the ability of a battery to perform its mission i.e. engage targets. This can possibly be measured by the number of targets not engaged out of the total targets available. This possibility has certain drawbacks. One is that the number of targets missed would be influenced as much by number of targets available as by the mix and ordering of warheads e.g., if fifty targets were presented, at least twenty-three would be missed (only twenty-seven missiles available); if 100 targets were presented, at least 73 targets or 73% would be missed, etc. This difficulty is avoided by considering the number of targets not engaged prior to the last missile being fired. This end point is used because restocking is not considered possible in a tactical time frame.

As was mentioned in the description of the system, only one target may be engaged at a time, regardless of the number of targets available. Hence, if a sufficient number of targets are available, ordering of warheads and reloading time are the governing factors of number of targets not engaged because no time would be available for reloading due to lack of targets. Actually, this condition can exist for analysis by assuming that the opponent is using saturation tactics. This is not a really gross assumption; it is merely assuming that the opponent is using optimal tactics [Waddell 1963]. This optimality is easily understood when

one considers the situation that would exist if saturation tactics were not used. If the targets were spread out in time and/or space, the battery could possibly be able to reload without paying a penalty in terms of inability to engage targets due to reloading.

A discussion of the role of time in this analysis is needed. When a section goes out of action in order to reload, the length of time that elapses until the section is again ready to fire has a direct effect upon the engagement capability of the battery. In order to make the transformation from distance each missile is moved during reload to time elapsed, it is assumed that the time required to move any missile to its adjacent position is constant; this constant is called one time unit. Hence to move a missile from storage #4 to launcher #2, five moves are required and consequently, five time units elapse. If, in addition, a move is made from storage #5 to launcher #3, five more time units would elapse or a total of ten time units would be required for the reload process.

The next problem is to relate the reload time to targets not engaged. As has already been explained, sufficient targets are available so that the battery can engage the next target immediately upon the detonation of the previous missile. Hence, the time between firings will be the flyout time plus the reaction time, which is assessment time (was the target destroyed?) plus decision time (reengage the same target? what warhead?). A common assumption in air

defense analysis is that the reaction time is constant [United States Army Air Defense School 1971] and was assumed to be so in this paper. Consideration of the fly-out time immediately reveals that this is a variable depending on speed of the missile, speed of the target, range of the target, path of the target, etc. If this fly-out time plus reaction time for some firing is longer than one time unit (as defined above), the battery could move one missile one place in each of the three sections without any decrease in engagement ability for it could not fire while the missile was in flight anyway. This would be an obvious benefit for the battery. However, virtually the only way to determine various fly-out times during a battle would be simulation due to the myriad parameters mentioned above. In this paper, it is assumed that the fly-out time is constant in order to enhance analysis.

Consideration of the magnitude of the constant fly-out time is necessary. Since both fly-out time and response time are constant, time between firings is now constant. If the time unit for reloading as defined above is less than this constant time between firing, the battery would be able to take advantage of the situation mentioned in the previous paragraph, i.e. movement of missiles with no penalty.

In the situation in which the reload time unit is greater than or equal to the inter-firing time, the equality case provides the lower limit of number of targets missed due to reloading. For instance, if the reload time unit is twice

as long as the inter-firing times, a reload requiring five time units would cause ten targets to be missed, while if the reload time unit is equal to the inter-firing time, only five targets would be missed during a reload of five time units. As a result, it is assumed that the reload time unit is equal to the inter-firing time. Consequently, this study yields a lower limit on targets missed due to reloading. This is the same as saying that this study provides an upper limit on engagement capability when faced with a saturation attack.

IV. THE MODEL

A. OPERATION PROCEDURES

Once a raid is detected, the battery commander selects a target, decides upon a suitable warhead, and orders the launching area to electrically prepare a missile containing the desired warhead for firing. At this point, the launching area either has the appropriate warhead available or not. Whether the desired one of the five possible types is in a position to be fired (on a launcher) depends upon the mix of the basic load, the order in which the basic load was stored, and the reloading policy. As was mentioned previously, the mix is assumed to be fixed by some external agency. The decision variable available to the battery is the order of storage of the warheads (which launcher or what position in storage would be the best location for each missile). This decision variable together with the reloading policies forms the battery's control of its ability to engage targets.

The question of optimal reloading doctrine is very difficult because at least two policies are possible. The first is to move only enough missiles to refill empty launchers. This allows short reload times at the beginning of the air battle. However, as the battle continues, reload times rapidly increase because the missiles have to be moved farther. For instance, suppose two missiles have been fired when a section enters reload status. In this case, the first and

second missiles in storage could be moved onto the launchers very quickly. Now suppose storage locations #1 to #4 are empty due to previous reloading. When reloading is again commenced, the missile in storage #5 (the closest available) obviously has to be moved across the four empty storage positions. Naturally this takes far longer than moving the missile in storage #2 onto a launcher.

The second possible policy is to move all missiles as far as possible each time the section reloads. This policy causes the longest reload times to be at the beginning of the air battle and hence, causes the reload times to decrease with each successive reload. For instance, suppose one missile has been fired when reload occurs. The missile in storage #1 would be moved onto the launchers, storage #2 would be moved to storage #1, etc., until all missiles in storage have been moved. In total, six missiles would be moved one position each. It should be noticed that at the finish of reloading, the empty storage location is at the far end of the storage barn i.e. storage #6 is empty. No missile is ever moved across empty storage locations under this policy.

The contrast between the two policies is easily discerned when a simpler system is considered, i.e. only one launcher and six storage locations. Under the first policy, the first reload takes one time unit (storage #1 moves to the launcher), the second reload takes two time units (storage #2 moves across storage #1 to the launcher) and so on. A

total of twenty-one time units would elapse during reloading. This is the sum: 1+2+3+4+5+6. Under the second policy, the first reload takes six time units (storage #1 moves to the launcher, storage #2 moves to storage #1, ..., storage #6 moves to storage #5). During the second reload, five time units would elapse (five missiles move one position) and so on. Thus, total time elapsed is 6+5+4+3+2+1=21. The difference in order of the two equal sums reflects the difference in the two policies. Because the first policy allows the shortest reload times at the beginning of the battle, it is assumed that this reloading policy would be followed. This has the intuitive appeal that during any attack of short duration the minimum time will be lost due to reload policy while during any attack of sufficiently long duration to exhaust all missiles in the section a decision maker would be indifferent between the two policies.

It should be noted that this policy assumes that each reload was a complete reload. For example, if two launchers were empty when the section stopped to reload, both launchers were filled. The difference in the two reloading policies resulted from movement inside the storage barn. It certainly is a feasible policy not to completely reload each time. However, the decision rules on when to and when not to reload completely are very tenuous and thus are extremely difficult, if not impossible, to model. Hence, the model in this study includes a complete reloading policy.

Additionally, another reloading policy has to be established; this policy concerns <u>when</u> the section reloads as opposed to <u>how</u> the section reloads as discussed above. The section has to reload at least twice: fire three missiles, reload, fire three missiles, reload. At the most the section would reload six times: fire first missile, reload, fire second missile, reload, . . . , fire sixth missile, reload.

A possible policy is to require a reload after a fixed number of missiles have been fired, e.g. always reload after two missiles have been fired. However, this policy ignores the purpose of the section i.e. provide warheads/missiles requested by the battery commander to engage the target, because the next request might have been for the last missile available. Rather than use such a policy, this study assumes a policy based upon the series of demands for warheads. The policy is to stop to reload whenever a demand cannot be satisfied exactly. This policy includes the feature that if the request for a warhead cannot be satisfied exactly, the request is shifted to another section. If the request cannot be satisfied in the battery, the target is allowed to pass unengaged. This assumption is not as reckless as it immediately appears. This type of missile unit is considered to be part of a defense-in-depth. Thus the fact that one battery does not engage a target does not imply that the target reaches its objective unscathed because batteries further behind could possibly engage it.

Of course, if the battery is not a part of a defense-in-depth or is the last battery in line, such a policy would be questionable. However, for this study it is assumed that this is a realistic procedure.

The exact response procedure requires some consideration. If the warheads can be placed in some order of effectiveness, such as size of explosive warhead, it is conceivable that a firing policy could exist in which a request for one type of warhead could be answered with that exact type or with a bigger type warhead. This policy would be a realization of a minimum-assurance-of-destruction philosophy. Unfortunately, further consideration makes it obvious that under such firing policy, the battery would soon be left with only small warheads. Additionally, it is conceivable that the warheads cannot be placed in an order of effectiveness. For instance, the warheads could be designed for such diverse objectives that they could have only a nominal rather than an ordinal ordering. As a result of these considerations as well as the defense-in-depth concept, it is assumed that the firing policy will be an exact response or none.

Given this exact response procedure, the reason for an either fire or reload policy becomes slightly more apparent. As was discussed previously, the interarrival times of targets are assumed to be a constant equal to one time unit where one time unit is the time required to move a missile over one position. Hence, if a request cannot be met, one time unit is immediately available for reloading.

Consequently, the policy in this study is to either fire or reload depending on whether a desired warhead is available or not.

Another reloading policy must be established. This policy concerns the decision as to which launcher would be designated to fill a request when two or more missiles with appropriate warheads are on the launchers. For example, if launchers #1 and #3 both contain the same type warhead and a request arrives for that type warhead, a decision has to be made concerning which launcher (#1 or #3) will be used to fill the request.

Intuitively, it would seem that launcher #3 is the best choice because, if the section stops to reload on the next request, launcher #3 can be more quickly refilled than if #1 is empty and the missiles on launchers #2 and #3 must both be moved over one position and then one missile moved out of storage. This procedure is used in the analysis. However, to be sure that intuition does indeed hold true, its validity is checked in the investigation portion of this thesis.

B. ANALOGY TO A MARKOV CHAIN

Since the interarrival rate of requests is a constant, the stream of requests forms a deterministic queue. As the requests arrive, they can either be filled or not filled. If it is filled, a missile is fired and a counter (the state variable) increases by one. If the request is not filled, reloading occurs and the state variable remains constant

for the number of time units required for the reloading because firing is impossible during reloading. This continues until the state variable reaches a value of nine because then the section would be empty. This procedure forms a pure birth process with a discrete time space, a special case of Markov Chains. The number of time units elapsed until the state variable reaches nine indicates the number of targets missed due to reloading because each time unit elapsed is equivalent to one target missed since it has been assumed that the inter-firing time is equal to the reloading time unit.

A modification of the Markov Chain analogy is necessary. Immediately after the finish of reloading, another request arrives. If this request cannot be filled, the time variable does not increase because no reloading is required (all launchers would be filled) and no targets are missed due to reloading. As a result, the time parameter increases after a reload are conditioned on the arrival of a request that could be filled. All intervening unfilled requests are ignored.

Continuing the Markov Chain analogy, this study is interested in the number of time increments elapsing from the beginning until the state parameter reaches nine. At least nine increments will pass because one missile is fired each time increment. As was mentioned above, at least two reloads will be necessary and as many as six may occur. Hence, the time parameter will be considerably larger than

nine. Actually, this study is concerned with the number of time increments during which the state parameter remains constant because all possible arrangements of warheads will require the nine time increments to fire the missiles. The objective is to discover if it is possible to affect significantly the number of time increments required and its resultant of missed targets due to reloading.

V. ANALYSIS

This analysis will concern a section consisting of three launchers and a storage barn for six missiles; thus nine missiles will be considered. Additionally, each missile is equipped with one of five possible warheads. Thus $x_2=3$ indicates that the missile in position #2 contains a type 3 warhead. For the rest of this discussion, the missile will be referred to as a type 3 missile since, even though the missiles are all constructed identically, the different warheads cause each to be used for different purposes.

There exist seven possible configurations of empty launchers when the section stops to reload. These seven possibilities come from the fact that position #1 or position #2 or position #3 will be empty where the "or" is the mathematical inclusive "or". Since there exist three positions that are empty or full, there are 2^3 or eight possible configurations of empty launchers. However, one possible configuration is that all launchers are full. This possibility has no physical meaning because the section will not stop to reload if it has not fired any missiles. An alternate and equally valid interpretation is that zero time will be required in order to reload. Either way, the configuration will not occur and hence, $2^3 - 1 = 7$ possible configurations will occur.

Now, the seven possible configurations are not equally likely. In fact, the probability of a certain configuration being realized is dependent upon what missiles (that is, warheads) are on the launchers and the probability that each type of warhead is requested. Y_k can be interpreted as the probability that a request will be for the warhead on launcher #k and equivalently, the probability that launcher #k is empty when the section stops to reload.

For an illustration, the probability that the second and third launchers are empty when the section stops to reload is

$$\frac{2y_2y_3(1 - y_1)}{y_1 + y_2 + y_3} \tag{1}$$

The numerator is the probability that two independent events occur and that the two can occur in two different orders. The factor in parenthesis is the probability of a failure on the third request; this is a necessary term to ensure that exactly two requests are satisfied. The denominator is the conditioning factor. The conditioning is the probability that one success has occurred. Hence, the total expression is the probability that launchers #2 and #3 are empty when the section reloads given that one request has been fired. This conditioning is necessary because any targets missed before the first firing are ignored.

During the reloading process, for this illustration, the missile in storage #1 is moved to launcher #2 (2 time

units required) and the missile in storage #2 is moved to launcher #3(2 time units required). By the model threat characteristics described previously, the four time units consumed result in four requests unfilled. The section is then able to resume answering requests because its launchers are filled — but possibly with warheads different from the original set; four requests have been missed; and storage locations #1 and #2 are empty. The probability that this situation is reached when the section is ready for the second round is given by Equation (1). The formulae for calculating the probability of all seven possible situations are given in Table I.

It must be noted that Equation (1) assumed that all warheads on the launchers are different. However, this is not certain. The formulae for calculating the probability of each configuration occurring in the four situations in which two or three of the three warheads are identical are given in Tables II, III, IV, and V. Note that in Tables I to V, the probability formulae sum to one. This is necessary since each table is an exhaustive list of all possible configurations when the section stops to reload. Additionally, Tables II to V indicate that some of the possible configurations of empty launchers have a probability of zero. This is due to the reloading doctrine requiring that when more than one launcher could fill a request, the one closest to the storage barns will be used.

In order to discover how many requests are not filled, it is only necessary to sum the number of requests that are missed each time the section reloads until the storage barn is emptied — for then the section would no longer stop to reload. The probability of missing that sum would be the probability of realizing that particular series of empty launcher configurations.

As an illustration, consider a basic load of warheads emplaced in the following manner: {1 2 3 4 5 1 2 3 4} where {1 2 3} are on the launchers and {4 5 1 2 3 4} are in the storage barn. Suppose all three launchers are emptied. The probability of this event is (from Table I since all warheads on launchers are different):

$$\frac{6y_1y_2y_3}{y_1 + y_2 + y_3} \quad . \tag{2}$$

During this reload, nine time units would be required (storage missile #4 to launcher #1, ... , storage missile #6 to launcher #3) and consequently, nine requests are unfilled. The section is then ready to fire again. Suppose all three launchers are emptied again. This reloading will cause eighteen requests to be unfilled and the probability of this happening is Equation (2) again. Hence, the probability of twenty-seven (nine plus eighteen) requests being unfilled is:

$$\left(\frac{6y_1y_2y_3}{y_1 + y_2 + y_3}\right)^2 \quad . \tag{3}$$

This procedure can be followed for all possible sequences of reloading by keeping track of requests missed and multiplying appropriate probabilities as required from Tables I to V.

Unfortunately, there exist a very large number of paths in the resulting decision tree. This is evident when it is considered that each node can have up to seven paths leading from it. It is true that one branch ends after two nodes (the one described immediately above), but other branches would continue for six nodes (only one missile fired each time the section stopped to reload). Consequently, an evaluation of the distribution of targets missed has to be accomplished by either computer calculation of the probabilities in the decision tree or computer simulation of the system.

In order to have some means of judging the validity of the computer calculations (see Computer Program I), three criteria are used. The first criterion is that the calculated mass function for number of requests not filled must sum to one. This is not as trivial as it sounds. The program is designed to feature separate calculations for each possible branch and then add the calculated branch mass to the mass of the appropriate point in the domain of the missed request random variable. If, after all possible branch masses have been calculated, the sum is indeed one, it can be felt that the program covers all possible branches. This indirect type of assurance of validity is needed for such a

program because it is impossible to manually verify the results. Additionally, the computer solution can keep track of the number of branches that yield each particular value of targets missed.

Another criterion established to judge the validity of the computer solution involves the minimum possible results. Examination of the reloading system reveals a situation that appears to yield the minimum number of targets missed. The situation is that only one launcher is empty each time reloading is commenced and that this launcher is the one closest to the storage barn. In other words, six reloads are required, each one with launcher #3 empty. The total requests missed in this case is twenty-one. In general, for a section configured such as the one in this study and with this measure of effectiveness, the minimum number of failures to supply requested warheads will be:

$$M_{min} = \sum_{i=1}^{S} i \quad . \tag{4}$$

The last criterion concerns the maximum number of requests missed. This arises when only one missile has been fired between each reload but launcher #1 is empty rather than launcher #3. In other words, instead of the empty launcher being the closest to storage, the empty space is the farthest from storage. The first reload would cause three requests to be missed because launcher #2 must be moved to launcher #1, launcher #3 then moved to launcher #2,

and finally storage #1 moved to launcher #3. In similar fashion, the second reload causes four requests to be missed until the sixth and last reload causes eight requests to be missed.

In summary, the total number of targets missed will be

$$\sum_{i=3}^{8} i = \sum_{i=1}^{8} i - \sum_{i=1}^{2} i \quad .$$

The right hand side has the intuitive content that the first term predicts the number of unfilled requests if the system consisted of one launcher and a storage of eight. This is completely analogous to Equation (4). But since this was not a single launcher system, the second term is a correction term for the intervening launchers. The second term is the number of requests missed if the missiles on launchers #2 and #3 would indeed have to be moved to launcher #1. Since these two missiles do not move in such fashion, the sum is subtracted from the total. This result is easily expanded to the general case:

$$M_{max} = \sum_{i=1}^{S+L-1} i - \sum_{i=1}^{L-1} i \quad . \tag{6}$$

The analytical program was run on several test cases utilizing Tables I to V. In all cases the calculated mass function summed to a number larger than .9999 and had a domain from 21 to 33 (the limits predicted by Equations (4)

and (6)). Hence, it appeared that the program was indeed performing properly and thus could be used to calculate the mass functions of number of targets for any possible permutation of warheads and probabilities of requests for each type of warhead.

During conduct of this trial phase, one surprising result was noticed. The minimum number of targets missed in all cases was twenty-one as predicted. However, the output of the program indicated that this result was achieved in as many as four ways instead of only one way (the one used to establish Equation (4)). It was not possible to easily find the alternate paths to this minimum result because as many as 4529 paths existed in some cases.

VI. SIMULATION

A method of validating computer solutions is to find the solution by another method. In this situation, computer simulation is a convenient alternative. Accordingly, a simulation was constructed embodying all the characteristics discussed previously (see Computer Program II). A random number generator was included to produce a series of requests of warheads in accordance with the p_j. With the request, the computer then determined if a matching warhead was on the launchers. If so, the position (x_i) was set to a value of zero and another request was generated. If not, reloading was accomplished, a record of requests missed during reloading was updated, and then the next request was generated. This procedure continued until the storage barn was emptied. Then a counter corresponding to the number of requests missed was increased (if 27 requests were missed, the counter for event "27" was increased by one), the section was reinitialized to the original basic load of warheads, and a new stream of requests was presented to the section. This procedure was performed for a fixed number of repetitions and then an observed mass function was calculated by dividing the observed occurrence of each event, e.g. total number of times that 23 requests were missed, by the total number of repetitions. Additionally, records were maintained of the requests in order to ensure the proper proportion of types of requests were indeed being presented to the section.

Table VI presents the simulated mass function of the random variable, requests missed, for varying numbers of repetitions. It will be noticed that a very large number of repetitions was necessary in order to produce a steady mass function.

Using the Kolmogorov-Smirnoff test to determine if the simulation results from a sample of 15,000 could have been realized from the analytical mass function [Siegel 1956], it was determined that the mass function yielded by the simulation did indeed come from the analytical mass function, that is, the two were identically distributed. The size of this hypothesis test was .05 [Zehna 1970]. In other words, there existed a 5% probability that the test could have declared the two mass functions different when in fact they were identical (See Table VII). This result was very encouraging since the author had very high confidence that the simulation accurately performed in the manner of the modeled system. With the hypothesis test revealing that the simulation and the analytical model produced the same results, it was felt that the analytical work was accurate.

VII. INVESTIGATION OF THE SYSTEM'S CHARACTERISTICS

Since the number of requests missed due to reloading time is a random variable, some means are necessary to determine if one mass function is preferable to another mass function. The different mass functions would result from different ordering of the warheads in the section.

The most familiar criterion and easiest to calculate and understand for ranking desirability of random variables is that of relative sizes of expected values. This criterion ranks mass functions in order of their expected values. In this study, the objective is to minimize requests missed due to reloading. Hence, the arrangement of warheads that produces the smallest expected value of the random variable would be the most desirable arrangement.

However, another more inclusive criterion is available. This is ordering by survivor functions. For a random variable M, the survivor function is defined as:

$$\overline{F}(m) \equiv \mathrm{Prob}\{M>m\} \qquad (7)$$

where m is an element of the domain of the random variable M. Note that the survivor function satisfies

$$\overline{F}(m) = 1.0 - F(m) \qquad (8)$$

where F(m) is the well known cumulative distribution function. For two random variables M_1 and M_2 (resulting from two warhead arrangements), it is true that if $\overline{F}_1(t) \geq \overline{F}_2(t)$ then

$E(M_1) \geq E(M_2)$. Hence, if while comparing two particular warhead arrangements, it could be shown that $\bar{F}_1(t) \geq \bar{F}_2(t)$, a considerably stronger ranking would be known because the expected value ranking would be identical. On the other hand, if it is shown that the two expected values are different, nothing is known (without further calculations) about the stronger ranking.

With this knowledge, it can then be determined what warhead orderings are preferable to others. For example, Tables VIII, IX, and X provide the expected values and survivor functions for three different arrangements of one mix of warheads. Examination reveals that $E_{10} < E_9 < E_8$ with E_9 and E_8 virtually equal. Also it is noted that $\bar{F}_{10} \leq \bar{F}_9$ and $\bar{F}_{10} \leq \bar{F}_8$ while $\bar{F}_9$ and $\bar{F}_8$ have no clear cut dominance. This can be expected since their expected values are so close to each other. Thus the warhead arrangement in Table X is preferable because its survivor function dominates the other two.

Examination of $\bar{F}_9$ reveals a discrepancy. $\bar{F}_9$ does not become .0000 at m=34 as necessary because m=33 was the largest value of m with a nonzero mass. The explanation is that the analytic program yields a mass function that does not sum exactly to one and the difference appears as the nonzero value of $\bar{F}_9$ at m=34.

At this point, it is apparent how the best warhead arrangement can be determined i.e. find the permutation of warheads that yields a mass function that dominates all

other possible mass functions. Unfortunately, there exist numerous possible permutations of any given mix of nine warheads. The upper limit of possible permutations would occur when the mix is 2:2:2:2:1. In this case, the number of permutations would be

$$\frac{9!}{(2!)(2!)(2!)(2!)(1!)} = 7560 \qquad (9)$$

[Hickman 1971]. It is immediately evident that the size of this task is overwhelming. If it was judged worthwhile, the optimal permutation could be obtained by using the analytical computer program to calculate the mass function for every one of the possible permutations and then use the two criteria described above to find the best permutation. It is certainly conceivable that a unique solution would not exist. However, the criteria would still be able to eliminate poor permutations until a set was delineated that could not be differentiated further. At this point, an arbitrary selection would be in order. It is noted that this process in essence eliminates poor choices and then selects a solution from the remaining possibilities instead of finding the optimal solution directly.

Since finding the set of "best" permutations involves evaluating all possible permutations, some faster method is desirable because the computer analytical evaluation of each mass function requires approximately four seconds. The upper limit on possible run times is then slightly under

eight and one-half hours! In an effort to determine characteristics of the arrangement in Table X which were not present in the arrangements in Tables VIII and IX, it was noticed that the warheads were placed in a nondecreasing order of likelihood of request, e.g. p_5 warheads were on the launchers while p_1 warheads were in the rear of the storage barn when $p_1 \geq p_2 \geq p_3 \geq p_4 \geq p_5$.

As a result, a factorial computer experiment was conducted. Two effects were considered: (1) the effects of the monotone arrangement of warheads, and (2) the effects of the firing policy of firing the launcher closest to the barn when two missiles on the launchers contained the desired warhead. The question concerning the latter effect was explained earlier. The experiment was conducted with the simulation since it had been shown to yield very accurate results and was easier to modify than the analytic program.

Comparison of parts A and B of Table XI revealed that the firing doctrine had virtually no effect on the performance of the section loaded in the manner shown. This seemed reasonable since the doctrine concerned situations in which two or three identical warheads were on the launchers. With the loading plan as shown, duplication of warheads would not occur very often and hence, the firing doctrine would have little opportunity to manifest itself.

Examination of Tables XII and XIII revealed that, for the permutations of warheads which caused frequent duplication of warheads on the launchers, the firing doctrine

under study resulted in a dominance by expected values and by survivor functions in both tables. Hence, the assumption previously made to use this doctrine appears to be reasonable.

With this doctrine established as beneficial, examination of part A of Tables XI, XII, and XIII revealed that Table XIII dominated both Tables XI and XII at the survivor function level. It was also indicated that no particular dominance existed between Tables XI and XII. Even though the expected value of Table XI was less than that in Table XII, the difference was so slight that any declaration of dominance would be reckless because simulation was used to obtain the results.

At this point it was again noticeable that the permutation of warheads with a nondecreasing ordering of warheads yielded the dominant mass function. Many more trials could have been conducted, but since so many orderings were possible (perhaps, it was not necessary for a nondecreasing ordering but merely an overall upward trend in probability) the procedure seemed to promise little additional knowledge in return for much more computer run time. In fact, several more trials were conducted with no results contradicting the principle delineated above.

Instead, an explanation was discovered for why the nondecreasing orderings appeared to be the rule of dominant permutations. Based upon the probability formulae in Tables I to V and the results of considering the firing doctrine

of closest-launcher versus farthest-launcher in case of duplication, it was discernable that, in terms of requests missed during reload, it was preferable to have the empty launchers closest to the storage barn whenever possible. Actually this is in keeping with judgement: if two launchers are empty, it is best that they are launchers #2 and #3 because missiles can be moved directly from storage to a launcher without losing time moving a missile between launchers. Subsequently, the question arose: given three warheads to be placed on the launchers, what is the best arrangement to ensure that launcher #3 would be empty more often than #2 and that #2 would be empty more often than #1? If the three warheads are identical, the firing doctrine could ensure the desired configuration; if the three warheads are not identical, it is obvious that the best arrangement is to put the warhead most likely to be requested on launcher #3, the warhead next most likely to be requested on launcher #2, etc.

When consideration is given to what warhead should be placed in storage #1 after the three on the launchers have been placed, it appears that the same principle applies. The warhead in storage #1 should have a p_j greater than or equal to y_3 because the warheads on the launchers have already been arranged so that launcher #3 would most likely be empty. With launcher #3 empty, the missile in storage #1 would be moved to the launcher. When this occurs, it would still be desired that $y_3 \geq y_2 \geq y_1$. This would be

accomplished when storage #1 has a p_j greater than or equal to y_3. This reasoning is then immediately extended to all positions in the storage barn. In conclusion, it can be established that the number of requests not filled due to reloading will be minimized by arranging the given assortment of warheads in an order beginning with the least requested warhead on launcher #1, an equally or more likely warhead on launcher #2, and so on until all nine warheads are positioned.

VIII. CONCLUSIONS AND RECOMMENDATIONS

At this point, the object of the study has been achieved: Specifically, it is known what principle of ordering warheads in a sequential storage system will result in the minimum number of unfilled requests. However, an examination of the principle reveals a startling conclusion: the best ordering for minimizing misses due to reloading causes the worst performance in terms of availability! This is immediately evident when the permutation in Table X is considered. With the reloading policies as specified in this study, no penalty is suffered for requests missed until the first success. The permutation enhanced the probability that this first success would be on launcher #3, a situation shown to be best. Then since the least likely requested warheads were on the launchers, it is most likely that the next request will not be filled and by policy, the section then stops to reload. It was indicated earlier that the sequence of firing once and then reloading led to the minimum number of misses during reloading, i.e. 21 misses.

But consideration has to be given to the number of misses until the first success after each reloading. This number is a random variable with a geometric distribution. The single parameter of this distribution is the probability of a success on a single request. In this case, the parameter is the probability that a request is for a warhead

on the launchers. However, the optimal ordering discovered earlier placed the least probable warheads on the launchers. Hence, the size of the parameter would be small. The consequence is that, since the expected value of a geometric distribution is the reciprocal of the parameter, the number of misses before the first success would be large. In summary, the ordering of warheads that minimizes number of requests missed due to reloading simultaneously maximizes the number of requests missed due to availability! The immediate conclusion is that the optimal ordering sequence is extremely sensitive to the objective function.

More formally, consider an objective function:

$$M_s = \sum_{i=1}^{T} (a_i + m_i) + a^f \quad . \tag{10}$$

This form of the objective function is more meaningful when the sequence of events in a section is considered. When the section begins the battle, a number of requests will be missed at first (a_1) because of the type of warheads available on the launchers. This number would be a geometric random variable whose parameter depends on y_1, y_2, and y_3. Then either one, two, or three requests will be filled after which the section will stop to reload. A certain number of requests would be missed during reloading (m_1). This number would be a random variable depending on which launchers were empty and which storage locations were filled. After reloading, requests would again be missed due to availability

(a_2). This variable would again be distributed geometrically but with a parameter different from the first. At least one success will be experienced and then reloading will again occur with m_2 requests missed. This will continue until the last reload (T) empties the storage barn. T is a discrete random variable with a domain of 2, 3, 4, 5, 6 (as was explained earlier). The mass function of this discrete random variable is not readily apparent. Finally, after the storage barn is empty, a certain number of requests will be missed until all warheads on the launchers are requested (a^f). This number will be similar to a negative binomial random variable with a parameter depending on the p_j of the last three warheads.

Rearrangement of (10) yields

$$M_s = \sum_{i=1}^{T} a_i + \sum_{i=1}^{T} m_i + a^f \quad . \tag{11}$$

where the first term is a random sum of a variety of possible geometric distributions, and the second term is a random sum of random misses and the last term is similar but not identical to a negative binomial distribution. Hence, M is the sum of three random variables each with a rather complex mass function. As a result, the mass function of M would be extremely difficult to develop analytically.

This study concentrated on the second term by setting the first and third terms to zero by the defense-in-depth capability. The analysis shown on Tables I to V provided

the mass function of the second term. The result was that the objective of this study was to minimize

$$B_s = \sum_{i=1}^{T} m_i \quad . \tag{12}$$

This was found to be accomplished by placing the least likely warheads on the launchers first. Examination of this result revealed that this policy would make

$$A_s = \sum_{i=1}^{T} a_i + a^f \tag{13}$$

a comparatively large number. Intuition based upon earlier comments indicated that if the objective had been to minimize A_s, the optimal policy would be the opposite of that found in the previous chapter, i.e. most likely warheads placed on the launchers and each warhead's location based on descending size of p_j. Further intuition also indicates that if the objective had been to minimize M_s, that is minimize the number of targets missed due to both reloading and availability, the optimal arrangement would be a compromise between the two extreme arrangements called for by minimizing either A_s or B_s individually. Of course, these two possible objectives should be subjected to formal analysis in order to determine if the insights gained in this thesis were sufficient for intuition to indeed provide the correct solutions.

After that is accomplished, investigation could be conducted on the optimal ordering for a battery. In this case

$$M_B = \sum_{j+1}^{3} m_s \tag{14}$$

since a battery consists of three sections. Substitution of (11) into (14) yields

$$\begin{aligned} M_B = A_B + B_B = & \left(\sum_{j=1}^{3} \sum_{i=1}^{T} a_{ji} + \sum_{j=1}^{3} a_j^f \right) \\ & + \sum_{j=1}^{3} \sum_{i=1}^{T} m_{ji} \quad . \end{aligned} \tag{15}$$

Analogous to the section, it could be assumed that missing a request due to availability does not incur any penalty because other batteries exist to handle targets missed. In which case, the objective would be to minimize B_B. Similarly, the objective could be to minimize A_B or to minimize M_B. At this level, suboptimization might also arise in that the policy that optimizes one section might not optimize the battery if all three sections were loaded in accordance with an optimal section plan. For example, as was mentioned above, a section ordering plan to minimize M_s would likely be a compromise between the extreme strategies demanded to minimize A_s or B_s. However, the optimal ordering for a M_B might consist of two sections ordered to minimize A_s

and one section ordered to minimize B_s rather than all three ordered to minimize M_s.

In a continuation of the structuring effort, an objective for an entire defense would be

$$M_D = \sum_{k=1}^{B} M_B = \sum_{k=1}^{B} (A_B + B_B)$$

$$= \sum_{k=1}^{B} \sum_{j=1}^{3} \sum_{i=1}^{T} a_{kji} + \sum_{k=1}^{B} \sum_{j=1}^{3} a_{kj}^{f} \tag{16}$$

$$+ \sum_{k=1}^{B} \sum_{j=1}^{3} \sum_{i=1}^{T} m_{kji}$$

At this level of aggregation all terms must be considered for there are no batteries behind the defense. Thus targets missed due to both availability and reload time must be considered. To reflect the relative importance of targets missed as the position of a battery in a defense varies, a weighting factor might be introduced. For instance, if a defense of five batteries is organized with two batteries on line followed by a third battery followed by the fourth and fifth batteries on line, targets missed due to availability by batteries #1 and #2 would have little consequence because a back up capability exists. However, batteries #4 and #5 must stop all targets. Hence, let q_k denote a weighting factor of a target missed by battery k. For example, a possible set of weighting factors is:

$$q_1 = q_2 = 1$$
$$q_3 = 3 \tag{17}$$
$$q_4 = q_5 = 10$$

These reflect rapidly increasing importance to a target being missed as it penetrates the defense.

Using this weighting factor in Equation (16), the objective function to be minimized is:

$$M_D = \sum_{k=1}^{B} \sum_{j=1}^{3} \sum_{i=1}^{T} (q_k a_{kji} + s_k m_{kji}) + \sum_{k=1}^{B} \sum_{j=1}^{3} q_k a^f_{kj} \tag{18}$$

where q_k and s_k allow for possibly different weighting factors for targets missed due to availability and for targets missed due to reloading.

Again it appeared that suboptimization could present a problem at this defense level. The optimal defense ordering of warheads may not be five (or however many batteries there are) optimal battery arrangements. In fact, if weighting factors are used, the optimal defense ordering policy almost certainly would not be five optimal battery arrangements.

Another possible form of an objective function involves a slightly different weighting value than those in (17). This method further places a weight upon what type of targets are missed by which battery. It was assumed that

five types of targets were possible. Hence, the objective function to be minimized would be:

$$M_D = \sum_{k=1}^{B} \sum_{j=1}^{3} \sum_{i=1}^{T} (q_{k\ell} a_{kji} + s_{k\ell} m_{kji})$$
$$+ \sum_{k=1}^{B} \sum_{j=1}^{3} q_{k\ell} a^{f}_{kj} \quad . \tag{19}$$

In this manner, the penalty for battery k missing a type ℓ target due to availability may be different from the same battery missing the same target due to reloading. Using misses due to reloading as an example, the weighting factor for a battery in the first echelon of defense missing a nuclear weapon carrier would be less than the weighting factor for a battery in the last echelon missing the same target. On the other hand, one of the five possible types of targets may be of such low priority that no difference in weighting factors between batteries is warranted. This last objective function provides enough flexibility for weightings for different missions and goals that virtually any situation could be modeled. Unfortunately, solutions of these possible objective functions would be quite complicated. However, based upon the results of the section analysis, it is apparent that suboptimization is a very real possibility in this type of system and consequently, any further analysis must be conducted at the defense-level if possible.

TABLE I: Probability Formulae for $y_1 \neq y_2$, $y_2 \neq y_3$, and $y_1 \neq y_3$

Position of Empty Launcher			Probability of Occurrence
1	2	3	
X			$\frac{y_1(1 - y_2 - y_3)}{y_1 + y_2 + y_3}$
	X		$\frac{y_2(1 - y_1 - y_3)}{y_1 + y_2 + y_3}$
		X	$\frac{y_3(1 - y_1 - y_2)}{y_1 + y_2 + y_3}$
X	X		$\frac{2y_1y_2(1 - y_3)}{y_1 + y_2 + y_3}$
X		X	$\frac{2y_1y_3(1 - y_2)}{y_1 + y_2 + y_3}$
	X	X	$\frac{2y_2y_3(1 - y_1)}{y_1 + y_2 + y_3}$
X	X	X	$\frac{6y_1y_2y_3}{y_1 + y_2 + y_3}$

TABLE II: Probability Formulae for $y_1 = y_3$, and $y_1 \neq y_2$

Position of Empty Launcher 1	2	3	Probability of Occurrence
X			0
	X		$\frac{y_2(1-y_1)}{y_1+y_2}$
		X	$\frac{y_1(1-y_1-y_2)}{y_1+y_2}$
X	X		0
X		X	$\frac{y_1^2(1-y_2)}{y_1+y_2}$
	X	X	$\frac{2y_1y_2(1-y_1)}{y_1+y_2}$
X	X	X	$\frac{3y_1^2y_2}{y_1+y_2}$

TABLE III: Probability Formulae for $y_2=y_3$ and $y_1 \neq y_2$

Position of Empty Launcher 1	2	3	Probability of Occurrence
X			$\frac{y_1(1-y_2)}{y_1+y_2}$
	X		0
		X	$\frac{y_2(1-y_1-y_2)}{y_1+y_2}$
X	X		0
X		X	$\frac{2y_1y_2(1-y_2)}{y_1+y_2}$
	X	X	$\frac{y_2^{2}(1-y_1)}{y_1+y_2}$
X	X	X	$\frac{3y_1y_2^{2}}{y_1+y_2}$

TABLE IV: Probability Formulae for $y_1=y_2$ and $y_1 \neq y_3$

Position of Empty Launcher 1	2	3	Probability of Occurrence
X			0
	X		$\frac{y_1(1-y_1-y_3)}{y_1+y_3}$
		X	$\frac{y_3(1-y_1)}{y_1+y_3}$
X	X		$\frac{y_1^2(1-y_3)}{y_1+y_3}$
X		X	0
	X	X	$\frac{2y_1y_3(1-y_1)}{y_1+y_3}$
X	X	X	$\frac{3y_1^2y_3}{y_1+y_3}$

TABLE V: Probability Formulae for $y_1=y_2=y_3$

Position of Empty Launcher 1	2	3	Probability of Occurrence
X			0
	X		0
		X	$1-y_1$
X	X		0
X		X	0
	X	X	$y_1(1-y_1)$
X	X	X	y_1^2

TABLE VI: Results of Simulation with Various Number of Replications

i	1	2	3	4	5	6	7	8	9
x_i	1	2	3	4	5	1	2	3	4

j	1	2	3	4	5
p_j	.40	.25	.20	.10	.05

Number of Replications	100	500	1000	5000	10,000	15,000
Number of Targets Missed						
20	.0000	.0000	.0000	.0000	.0000	.0000
21	.0000	.0000	.0000	.0000	.0000	.0000
22	.0000	.0000	.0000	.0000	.0004	.0004
23	.0100	.0100	.0100	.0050	.0052	.0046
24	.0400	.0580	.0650	.0654	.0656	.0683
25	.1000	.1120	.1400	.1220	.1205	.1191
26	.2700	.2460	.2230	.2358	.2363	.2333
27	.2400	.2620	.2500	.2512	.2509	.2540
28	.2400	.1980	.1870	.1964	.1954	.1923
29	.0700	.0900	.1010	.0954	.0970	.0979
30	.0300	.0200	.0190	.0232	.0230	.0243
31	.0000	.0040	.0040	.0054	.0053	.0052
32	.0000	.0000	.0010	.0020	.0003	.0005
33	.0000	.0000	.0000	.0000	.0001	.0001
34	.0000	.0000	.0000	.0000	.0000	.0000
Execution Time in seconds (approx.)	7	13	21	81	160	333

(each column is a simulated mass function)

TABLE VII: Kolmogorov-Smirnoff Test Validating the Simulation in One Case

H_o: the mass function of the simulation is the same as the analytical mass function

H_1: the mass function of the simulation is different from the analytical mass function

$\alpha = .05$ $\qquad d_{crit} = \dfrac{1.36}{\sqrt{15,000}} = .0111$

i	1	2	3	4	5	6	7	8	9
x_i	1	2	3	4	5	1	2	3	4

j	1	2	3	4	5
p_j	.40	.25	.20	.10	.05

x	$F_o(x)$	$S_{15,000}(x)$	$\lvert F_o(x) - S_{15,000}(x)\rvert$
20	.00000	.00000	.00000
21	.00000	.00000	.00000
22	.00032	.00040	.00008
23	.00417	.00500	.00083
24	.07241	.07327	.00086
25	.19041	.19240	.00199
26	.42086	.42573	.00487
27	.66976	.67973	.00997
28	.86760	.87206	.00446
29	.96676	.96992	.00316
30	.99409	.99419	.00010
31	.99925	.99939	.00014
32	.99987	.99992	.00015
33	.99990	.99999	.00009
34	1.00000	1.00000	.00000

Since $d_{max} = .00997 < .0111 = d_{crit}$, accept H_o.

TABLE VIII: Analytical Mass and Survivor Functions for a Permutation of Warheads with Decreasing Order of Probability of Request

i	1	2	3	4	5	6	7	8	9
x_i	1	1	2	2	3	3	4	4	5

j	1	2	3	4	5
p_j	.40	.25	.20	.10	.05

Number of Targets Missed	Mass	Survivor Function ($\bar{F}_8$)
20	.0000	1.0000
21	.0001	.9999
22	.0016	.9983
23	.0049	.9935
24	.0403	.9532
25	.1573	.7959
26	.1772	.6187
27	.3062	.3125
28	.1747	.1378
29	.1097	.0281
30	.0281	.0000
31	.0000	.0000
32	.0000	.0000
33	.0000	.0000
34	.0000	

Expected value (E_8) = 26.837

TABLE IX: Analytical Mass and Survivor Functions for a Permutation of Warheads with an Arbitrary Order of Probability of Request

i	1	2	3	4	5	6	7	8	9
x_i	1	2	3	4	5	1	2	3	4

j	1	2	3	4	5
p_j	.40	.25	.20	.10	.05

Number of Targets Missed	Mass	Survivor Function ($\overline{F}_9$)
20	.0000	1.0000
21	.0000	1.0000
22	.0003	.9997
23	.0038	.9958
24	.0682	.9276
25	.1180	.8096
26	.2304	.5791
27	.2489	.3302
28	.1978	.1324
29	.0992	.0332
30	.0273	.0059
31	.0052	.0007
32	.0006	.0001
33	.0000	.0001
34	.0000	.0001

Expected value (E_9) = 26.811

TABLE X: Analytical Mass and Survivor Functions for a Permutation of Warheads with an Increasing Order of Probability of Request

i	1	2	3	4	5	6	7	8	9
x_i	5	4	4	3	3	2	2	1	1

j	1	2	3	4	5
p_j	.40	.25	.20	.10	.05

Number of Targets Missed	Mass	Survivor Function ($\bar{F}_{10}$)
20	.0000	1.0000
21	.0318	.9682
22	.0843	.8838
23	.2156	.6682
24	.2125	.4557
25	.1818	.2739
26	.1433	.1306
27	.0868	.0438
28	.0326	.0112
29	.0104	.0008
30	.0008	.0000
31	.0000	.0000
32	.0000	.0000
33	.0000	.0000
34	.0000	.0000

Expected value (E_{10}) = 24.435

TABLE XI: Comparison of Close-Launcher versus Far-Launcher Firing Doctrine for a Permutation of Warheads with an Arbitrary Order of Probability of Request

i	1	2	3	4	5	6	7	8	9
x_i	1	2	3	4	5	1	2	3	4

j	1	2	3	4	5
p_j	.40	.25	.20	.10	.05

Part A: Close Launcher

Number of Targets Missed	Mass	Survivor Function
20	.0000	1.0000
21	.0000	1.0000
22	.0004	.9996
23	.0052	.9944
24	.0656	.9288
25	.1205	.8083
26	.2363	.5720
27	.2509	.3211
28	.1954	.1257
29	.0970	.0287
30	.0230	.0057
31	.0053	.0004
32	.0003	.0001
33	.0001	.0000
34	.0000	.0000

Expected value = 26.785

TABLE XI (continued)

Part B: Far-Launcher

Number of Targets Missed	Mass	Survivor Function
20	.0000	1.0000
21	.0000	1.0000
22	.0000	1.0000
23	.0002	.9998
24	.0501	.9497
25	.1026	.8471
26	.2406	.6065
27	.2767	.3298
28	.2033	.1265
29	.0978	.0287
30	.0230	.0057
31	.0053	.0004
32	.0003	.0001
33	.0001	.0000
34	.0000	.0000

Expected value = 26.894

TABLE XII: Comparison of Close-Launcher versus Far-Launcher Firing Doctrine for a Permutation of Warheads with a Decreasing Order of Probability of Request

i	1	2	3	4	5	6	7	8	9
x_i	1	1	2	2	3	3	4	4	5

j	1	2	3	4	5
p_j	.40	.25	.20	.10	.05

Part A: Close-Launcher

Number of Targets Missed	Mass	Survivor Function
20	.0000	1.0000
21	.0001	.9999
22	.0017	.9982
23	.0068	.9914
24	.0418	.9496
25	.1565	.7931
26	.1712	.6219
27	.3156	.3063
28	.1699	.1364
29	.1107	.0257
30	.0257	.0000
31	.0000	.0000
32	.0000	.0000
33	.0000	.0000
34	.0000	.0000

Expected value = 26.822

TABLE XII (continued)

Part B: Far-Launcher

Number of Targets Missed	Mass	Survivor Function
20	.0000	1.0000
21	.0001	.9999
22	.0004	.9995
23	.0005	.9990
24	.0149	.9841
25	.0262	.9579
26	.1014	.8565
27	.1599	.6966
28	.2489	.4477
29	.2069	.2408
30	.1509	.0899
31	.0642	.0257
32	.0234	.0023
33	.0023	.0000
34	.0000	.0000

Expected value = 28.300

TABLE XIII: Comparison of Close-Launcher versus Far-Launcher Firing Doctrine for a Permutation of Warheads with an Increasing Order of Probability Request

i	1	2	3	4	5	6	7	8	9
x_i	5	4	4	3	3	2	2	1	1

j	1	2	3	4	5
p_j	.40	.25	.20	.10	.05

Part A: Close-Launcher

Number of Targets Missed	Mass	Survivor Function
20	.0000	1.0000
21	.0309	.9691
22	.0848	.8843
23	.2146	.6697
24	.2093	.4604
25	.1803	.2801
26	.1467	.1334
27	.0896	.0438
28	.0318	.0120
29	.0113	.0007
30	.0007	.0000
31	.0000	.0000
32	.0000	.0000
33	.0000	.0000
34	.0000	.0000

Expected value = 24.435

TABLE XIII: (Continued)

Part B: Far-Launcher

Number of Targets Missed	Mass	Survivor Function
20	.0000	1.0000
21	.0000	1.0000
22	.0309	.9691
23	.0932	.8759
24	.1356	.7403
25	.1835	.5568
26	.1974	.3594
27	.1503	.2091
28	.1128	.0963
29	.0610	.0353
30	.0239	.0114
31	.0100	.0014
32	.0014	.0000
33	.0000	.0000
34	.0000	.0000

Expected value = 25.855

COMPUTER PROGRAM I

```
      DIMENSICN THREAT(5),MISSIL(9),FACT(3),STORA(27C0,11),S
     CTCRB(2700,11),PROB(40),MOVES(40)
      DATA STORA/29700*0.0/,STORB/29700*0.0/,PRCB/40*0.0/,MO
     CVES/40*0/
      STCRB(1,10)=1.0
C
C     READ IN THE INITIAL ARRAY OF NINE WARHEADS.
C
      READ (5,8010) (STORB(1,J),J=1,9)
 8010 FCRMAT (9F2.0)
C
C     READ IN THE PROBABILITIES OF THE FIVE TYPES CF
C     WARHEAD REQUESTS.  THE FIVE PROBABILITIES MUST ADD TO
C     ONE.
C
      READ (5,8030) THREAT
 8030 FORMAT (5F4.3)
C
C     THIS OUTPUT ENSURES THAT THE INITIAL WARHEAC ARRAY
C     WAS READ IN PROPERLY.
C
      WRITE (6,8040) (STORB(1,J),J=1,9)
 8040 FORMAT ('0','FIRST ARRAY IS',9F3.0)
C
C     THIS OUTPUT ENSURES THAT THE WARHEAD REQUEST
C     PROBABILITIES WERE INPUT CORRECTLY.
C
      WRITE (6,8050) THREAT
 8050 FCRMAT ('0','WARHEAD PROBABILITIES ARE',5F1C.3)
C
C     THIS STEP DETERMINES WHEN ALL BRANCHES OF THE
C     DECISION TREE HAVE BEEN DELINEATED.
C
 2000 IF (STORB(1,9).EQ.0.0) GO TO 6000
C
C     THIS STEP LOADS THE RESULTING ARRAYS FROM THE
C     PREVIOUS CALCULATICNS INTO THE WORKING AREA.
C
      DC 3 JJK=1,2700
      DO 2 JJJ=1,11
      STCRA(JJK,JJJ)=STCRB(JJK,JJJ)
    2 CCNTINUE
    3 CCNTINUE
C
C     THIS STEP RESETS THE STCRAGE AREA TC ZERCS.
C
      DC 5 JJI=1,2700
      DO 4 JJL=1,11
      STCRB(JJI,JJL)=0.0
    4 CCNTINUE
    5 CONTINUE
      NB=1
      DO 5000 NNN=1,2700
C
C     THIS STEP DETERMINES IF ALL ARRAYS IN THE PRESENT
C     CALCULATIONS HAVE BEEN PROCESSED.
C
      IF (STORA(NNN,1).EQ.0.0) GO TO 2000

C     THIS STEP DETERMINES THE PROBABILITY FACTCRS
C     NECESSARY FOR THE CALCULATICNS DICTATED BY TABLES
C     I THROUGH V.
C
      DC 30 J=1,3
      DC 20 K=1,5
```

```
      IF (STORA(NNN,J).EQ.FLOAT(K)) FACT(J)=THREAT(K)
   20 CONTINUE
   30 CONTINUE
      DO 40 I=1,9
      MISSIL(I)=STORA(NNN,I)
   40 CONTINUE
      IF ((MISSIL(1).EQ.MISSIL(2)).AND.(MISSIL(2).EQ.MISSIL(
     C3))) GO TO 500
      IF (MISSIL(1).EQ.MISSIL(2)) GO TO 390
      IF (MISSIL(2).EQ.MISSIL(3)) GO TO 270
      IF (MISSIL(1).EQ.MISSIL(3)) GO TO 150
C
C     THE FORMULAE IN TABLE I ARE ENCODED BETWEEN HERE
C     AND LINE 150.
C
      DEN=FACT(1)+FACT(2)+FACT(3)
      MISSIL(1)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 53
      DO 51 N=1,9
      STORB(NB,N)=MISSIL(N)
   51 CONTINUE
      STORB(NB,10)=FACT(1)*(1.0-FACT(3)-FACT(2))*STORA(NNN,1
     C0)/DEN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 55
   53 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(1)*(1.0-FACT(3)-FACT(2))*STORA(NNN,10)/DEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
   55 DO 57 K=1,9
      MISSIL(K)=STORA(NNN,K)
   57 CONTINUE
      MISSIL(2)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 61
      DO 59 M=1,9
      STORB(NB,M)=MISSIL(M)
   59 CONTINUE
      STORB(NB,10)=FACT(2)*(1.0-FACT(1)-FACT(3))*STORA(NNN,1
     C0)/DEN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 63
   61 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(2)*(1.0-FACT(1)-FACT(3))*STORA(NNN,10)/DEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
   63 DO 65 NNM=1,9
      MISSIL(NNM)=STORA(NNN,NNM)
   65 CONTINUE
      MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 69
      DO 67 IN=1,9
      STORB(NB,IN)=MISSIL(IN)
   67 CONTINUE
      STORB(NB,10)=FACT(3)*(1.0-FACT(1)-FACT(2))*STORA(NNN,1
     C0)/DEN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 70
   69 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(3)*(1.0-FACT(1)-FACT(2))*STORA(NNN,10)/DEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=SPROB+PROB(ILDS)
   70 DO 71 LI=1,9
      MISSIL(LI)=STORA(NNN,LI)
```

```
71 CONTINUE
   MISSIL(2)=0
   MISSIL(3)=0
   NCRIT=0
   CALL RELOAD (MISSIL,NCRIT)
   IF (MISSIL(9).EQ.0) GO TO 75
   DO 73 LN=1,9
   STORB(NB,LN)=MISSIL(LN)
73 CONTINUE
   STORB(NB,10)=2.0*FACT(2)*FACT(3)*(1.0-FACT(1))*STORA(N
  CNN,10)/DEN
   STORB(NB,11)=STORA(NNN,11)+NCRIT
   NB=NB+1
   GO TO 77
75 ILDS=STORA(NNN,11)+NCRIT
   SPROB=2.0*FACT(2)*FACT(3)*(1.0-FACT(1))*STORA(NNN,10)/
  CDEN
   MOVES(ILDS)=MOVES(ILDS)+1
   PROB(ILDS)=SPROB+PROB(ILDS)
77 DO 79 LZ=1,9
   MISSIL(LZ)=STORA(NNN,LZ)
79 CONTINUE
   MISSIL(1)=0
   MISSIL(3)=0
   NCRIT=0
   CALL RELOAD (MISSIL,NCRIT)
   IF (MISSIL(9).EQ.0) GO TO 83
   DO 81 LA=1,9
   STORB(NB,LA)=MISSIL(LA)
81 CONTINUE
   STORB(NB,10)=2.0*FACT(1)*FACT(3)*(1.0-FACT(2))*STORA(N
  CNN,10)/DEN
   STORB(NB,11)=STORA(NNN,11)+NCRIT
   NB=NB+1
   GO TO 85
83 ILDS=STORA(NNN,11)+NCRIT
   SPROB=2.0*FACT(1)*FACT(3)*(1.0-FACT(2))*STORA(NNN,10)/
  CDEN
   MOVES(ILDS)=MOVES(ILDS)+1
   PROB(ILDS)=SPROB+PROB(ILDS)
85 DO 87 LB=1,9
   MISSIL(LB)=STORA(NNN,LB)
87 CONTINUE
   MISSIL(1)=0
   MISSIL(2)=0
   NCRIT=0
   CALL RELOAD (MISSIL,NCRIT)
   IF (MISSIL(9).EQ.0) GO TO 91
   DO 89 LC=1,9
   STORB(NB,LC)=MISSIL(LC)
89 CONTINUE
   STORB(NB,10)=2.0*FACT(1)*FACT(2)*(1.0-FACT(3))*STORA(N
  CNN,10)/DEN
   STORB(NB,11)=STORA(NNN,11)+NCRIT
   NB=NB+1
   GO TO 93
91 ILDS=STORA(NNN,11)+NCRIT
   SPROB=2.0*FACT(1)*FACT(2)*(1.0-FACT(3))*STORA(NNN,10)/
  CDEN
   MOVES(ILDS)=MOVES(ILDS)+1
   PROB(ILDS)=PROB(ILDS)+SPROB
93 DO 95 LD=1,9
   MISSIL(LD)=STORA(NNN,LD)
95 CONTINUE
   MISSIL(1)=0
   MISSIL(2)=0
   MISSIL(3)=0
   NCRIT=0
   CALL RELOAD (MISSIL,NCRIT)
   IF (MISSIL(9).EQ.0) GO TO 99
   DO 97 LE=1,9
   STORB(NB,LE)=MISSIL(LE)
```

```
   97 CONTINUE
      STORB(NB,10)=6.0*FACT(1)*FACT(2)*FACT(3)*STORA(NNN,10)
     C/DEN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 5000
   99 ILDS=STORA(NNN,11)+NCRIT
      SPROB=6.0*FACT(1)*FACT(2)*FACT(3)*STORA(NNN,10)/DEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
      GO TO 5000
C
C     THE FORMULAE IN TABLE II ARE ENCODED BETWEEN
C     HERE AND LINE 270.
C
  150 DEN=FACT(1)+FACT(2)
      MISSIL(2)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 154
      DO 152 NA=1,9
      STORB(NB,NA)=MISSIL(NA)
  152 CONTINUE
      STORB(NB,10)=FACT(2)*(1.0-FACT(1))*STORA(NNN,10)/DEN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 156
  154 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(2)*(1.0-FACT(1))*STORA(NNN,10)/DEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
  156 DO 158 KB=1,9
      MISSIL(KB)=STORA(NNN,KB)
  158 CONTINUE
      MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 162
      DO 160 KC=1,9
      STORB(NB,KC)=MISSIL(KC)
  160 CONTINUE
      STORB(NB,10)=FACT(1)*(1.0-FACT(1)-FACT(2))*STORA(NNN,1
     C0)/DEN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 164
  162 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(1)*(1.0-FACT(1)-FACT(2))*STORA(NNN,10)/DEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=SPROB+PROB(ILDS)
  164 DO 166 KD=1,9
      MISSIL(KD)=STORA(NNN,KD)
  166 CONTINUE
      MISSIL(1)=0
      MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 170
      DO 168 KE=1,9
      STORB(NB,KE)=MISSIL(KE)
  168 CONTINUE
      STORB(NB,10)=FACT(1)**2*(1.0-FACT(2))*STORA(NNN,10)/DE
     CN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 172
  170 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(1)**2*(1.0-FACT(2))*STORA(NNN,10)/DEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
  172 DO 174 KF=1,9
      MISSIL(KF)=STORA(NNN,KF)
```

```
174 CONTINUE
    MISSIL(2)=0
    MISSIL(3)=0
    NCRIT=0
    CALL RELOAD  (MISSIL,NCRIT)
    IF (MISSIL(9).EQ.0) GO TO 178
    DO 176 KG=1,9
    STORB(NB,KG)=MISSIL(KG)
176 CONTINUE
    STORB(NB,10)=2.0*FACT(1)*FACT(2)*(1.0-FACT(1))*STORA(N
   CNN,10)/DEN
    STORB(NB,11)=STORA(NNN,11)+NCRIT
    NB=NB+1
    GO TO 180
178 ILDS=STORA(NNN,11)+NCRIT
    SPROB=2.0*FACT(1)*FACT(2)*(1.0-FACT(1))*STORA(NNN,10)/
   CDEN
    MOVES(ILDS)=MOVES(ILDS)+1
    PROB(ILDS)=PROB(ILDS)+SPROB
180 DO 182 KH=1,9
    MISSIL(KH)=STORA(NNN,KH)
182 CONTINUE
    MISSIL(1)=0
    MISSIL(2)=0
    MISSIL(3)=0
    NCRIT=0
    CALL RELOAD (MISSIL,NCRIT)
    IF (MISSIL(9).EQ.0) GO TO 186
    DO 184 KI=1,9
    STORB(NB,KI)=MISSIL(KI)
184 CONTINUE
    STORB(NB,10)=3.0*FACT(1)**2*FACT(2)*STORA(NNN,10)/DEN
    STORB(NB,11)=STORA(NNN,11)+NCRIT
    NB=NB+1
    GO TO 5000
186 ILDS=STORA(NNN,11)+NCRIT
    SPROB=3.0*FACT(1)**2*FACT(2)*STORA(NNN,10)/DEN
    MOVES(ILDS)=MOVES(ILDS)+1
    PROB(ILDS)=PROB(ILDS)+SPROB
    GO TO 5000
C
C   THE FORMULAE IN TABLE III ARE ENCODED BETWEEN
C   HERE AND LINE 390.
C
270 DEN=FACT(1)+FACT(2)
    MISSIL(1)=0
    NCRIT=0
    CALL RELOAD (MISSIL,NCRIT)
    IF (MISSIL(9).EQ.0) GO TO 274
    DO 272 IA=1,9
    STORB(NB,IA)=MISSIL(IA)
272 CONTINUE
    STORB(NB,10)=FACT(1)*(1.0-FACT(2))*STORA(NNN,10)/DEN
    STORB(NB,11)=STORA(NNN,11)+NCRIT
    NB=NB+1
    GO TO 276
274 ILDS=STORA(NNN,11)+NCRIT
    SPROB=FACT(1)*(1.0-FACT(2))*STORA(NNN,10)/DEN
    MOVES(ILDS)=MOVES(ILDS)+1
    PROB(ILDS)=PROB(ILDS)+SPROB
276 DO 278 IB=1,9
    MISSIL(IB)=STORA(NNN,IB)
278 CONTINUE
    MISSIL(3)=0
    NCRIT=0
    CALL RELOAD (MISSIL,NCRIT)
    IF (MISSIL(9).EQ.0) GO TO 282
    DO 280 IC=1,9
    STORB(NB,IC)=MISSIL(IC)
280 CONTINUE
    STORB(NB,10)=FACT(2)*(1.0-FACT(1)-FACT(2))*STORA(NNN,1
   C0)/DEN
```

```
      STCRB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GC TO 284
  282 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(2)*( 1.0-FACT(1)-FACT(2))*STORA(NNN,10)/DEN
      MCVES(ILDS)=MOVES(ILDS)+1
      PROB(ILCS)=PROB(ILDS)+SPROB
  284 DC 286 IE=1,9
      MISSIL(IE)=STORA(NNN,IE)
  286 CCNTINUE
      MISSIL(1)=0
      MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 290
      DC 288 IF=1,9
      STORB(NB,IF)=MISSIL(IF)
  288 CCNTINUE
      STCRB(NB,10)=2.0*FACT(1)*FACT(2)*(1.0-FACT(2))*STORA(N
     CNN,10)/DEN
      STCRB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GC TO 292
  290 ILCS=STCRA(NNN,11)+NCRIT
      SPROB=2.0*FACT(1)*FACT(2)*(1.0-FACT(2))*STCRA(NNN,10)/
     CDEN
      MCVES(ILDS)=MOVES(ILDS)+1
      PRCB(ILCS)=PROB(ILCS)+SPROB
  292 DC 294 IG=1,9
      MISSIL(IG)=STORA(NNN,IG)
  294 CCNTINUE
      MISSIL(2)=0
      MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 298
      DC 296 IH=1,9
      STCRB(NB,IH)=MISSIL(IH)
  296 CCNTINUE
      STCRB(NB,10)=FACT(2)**2*(1.0-FACT(1))*STCRA(NNN,10)/DE
     CN
      STCRB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GC TO 300
  298 ILCS=STORA(NNN,11)+NCRIT
      SPROB=FACT(2)**2*(1.0-FACT(1))*STORA(NNN,10)/DEN
      MCVES(ILDS)=MCVES(ILDS)+1
      PROB(ILCS)=PROB(ILDS)+SPROB
  300 DC 302 IQ=1,9
      MISSIL(IQ)=STORA(NNN,IQ)
  302 CONTINUE
      MISSIL(1)=0
      MISSIL(2)=0
      MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 306
      DC 304 IR=1,9
      STCRB(NB,IR)=MISSIL(IR)
  304 CCNTINUE
      STCRB(NB,10)=3.0*FACT(1)*FACT(2)**2*STORA(NNN,10)/DEN
      STCRB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GC TO 5000
  306 ILCS=STORA(NNN,11)+NCRIT
      SPROB=3.0*FACT(1)*FACT(2)**2*STORA(NNN,10)/DEN
      MCVES(ILDS)=MCVES(ILDS)+1
      PRCB(ILCS)=PROB(ILCS)+SPROB
      GC TO 5000
C
C     THE FORMULAE IN TABLE IV ARE ENCCDED BETWEEN
C     HERE AND LINE 500.
```

```
C
  390 DEN=FACT(1)+FACT(3)
      MISSIL(2)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 394
      DO 392 MA=1,9
      STORB(NB,MA)=MISSIL(MA)
  392 CONTINUE
      STORB(NB,10)=FACT(1)*(1.0-FACT(1)-FACT(3))*STORA(NNN,1
     C0)/DEN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 396
  394 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(1)*(1.0-FACT(1)-FACT(3))*STORA(NNN,10)/DEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
  396 DO 398 IC=1,9
      MISSIL(IC)=STORA(NNN,IC)
  398 CONTINUE
      MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 402
      DO 400 ID=1,9
      STORB(NB,ID)=MISSIL(ID)
  400 CONTINUE
      STORB(NB,10)=FACT(3)*(1.0-FACT(1))*STORA(NNN,10)/DEN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 404
  402 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(3)*(1.0-FACT(1))*STORA(NNN,10)/DEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
  404 DO 406 IE=1,9
      MISSIL(IE)=STORA(NNN,IE)
  406 CONTINUE
      MISSIL(1)=0
      MISSIL(2)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 410
      DO 408 IF=1,9
      STORB(NB,IF)=MISSIL(IF)
  408 CONTINUE
      STORB(NB,10)=FACT(1)**2*(1.0-FACT(3))*STORA(NNN,10)/DE
     CN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 412
  410 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(1)**2*(1.0-FACT(3))*STORA(NNN,10)/DEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
  412 DO 414 IG=1,9
      MISSIL(IG)=STORA(NNN,IG)
  414 CONTINUE
      MISSIL(2)=0
      MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 418
      DO 416 IH=1,9
      STORB(NB,IH)=MISSIL(IH)
  416 CONTINUE
      STORB(NB,10)=2.0*FACT(1)*FACT(3)*(1.0-FACT(1))*STORA(N
     CNN,10)/DEN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 420
```

```
  418 ILDS=STORA(NNN,11)+NCRIT
      SPROB=2.0*FACT(1)*FACT(3)*(1.0-FACT(1))*STORA(NNN,10)/
     CDEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
  420 DO 422 IJ=1,9
      MISSIL(IJ)=STORA(NNN,IJ)
  422 CONTINUE
      MISSIL(1)=0
      MISSIL(2)=0
      MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 426
      DO 424 IK=1,9
      STORB(NB,IK)=MISSIL(IK)
  424 CONTINUE
      STORB(NB,10)=3.0*FACT(1)**2*FACT(3)*STORA(NNN,10)/DEN
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 5000
  426 ILDS=STORA(NNN,11)+NCRIT
      SPROB=3.0*FACT(1)**2*FACT(3)*STORA(NNN,10)/DEN
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
      GO TO 5000
C
C     THE FORMULAE IN TABLE V ARE ENCODED BETWEEN
C     HERE AND LINE 5000.
C
  500 MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 504
      DO 502 JQ=1,9
      STORB(NB,JQ)=MISSIL(JQ)
  502 CONTINUE
      STORB(NB,10)=(1.0-FACT(1))*STORA(NNN,10)
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 506
  504 ILDS=STORA(NNN,11)+NCRIT
      SPROB=(1.0-FACT(1))*STORA(NNN,10)
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
  506 DO 508 JR=1,9
      MISSIL(JR)=STORA(NNN,JR)
  508 CONTINUE
      MISSIL(2)=0
      MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
      IF (MISSIL(9).EQ.0) GO TO 512
      DO 510 JS=1,9
      STORB(NB,JS)=MISSIL(JS)
  510 CONTINUE
      STORB(NB,10)=FACT(1)*(1.0-FACT(1))*STORA(NNN,10)
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 514
  512 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(1)*(1.0-FACT(1))*STORA(NNN,10)
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
  514 DO 516 JT=1,9
      MISSIL(JT)=STORA(NNN,JT)
  516 CONTINUE
      MISSIL(1)=0
      MISSIL(2)=0
      MISSIL(3)=0
      NCRIT=0
      CALL RELOAD (MISSIL,NCRIT)
```

```
      IF (MISSIL(9).EQ.0) GO TO 520
      DO 518 JU=1,9
      STORB(NB,JU)=MISSIL(JU)
  518 CONTINUE
      STORB(NB,10)=FACT(1)**2*STORA(NNN,10)
      STORB(NB,11)=STORA(NNN,11)+NCRIT
      NB=NB+1
      GO TO 5000
  520 ILDS=STORA(NNN,11)+NCRIT
      SPROB=FACT(1)**2*STORA(NNN,10)
      MOVES(ILDS)=MOVES(ILDS)+1
      PROB(ILDS)=PROB(ILDS)+SPROB
      GO TO 5000
 5000 CONTINUE
C
C     THIS OUTPUT GIVES THE MASS FUNCTION OF THE NUMBER
C     OF MOVES REQUIRED TO EMPTY THE STORAGE BARNS AND
C     ALSO THE NUMBER OF BRANCHES YIELDING EACH PARTICULAR
C     VALUE OF NUMBER OF MOVES.
C
 6000 WRITE (6,8095)
 8095 FORMAT ('0','THE PROBABILITIES OF MOVES ARE  THE OCCUR
     CRENCES OF MOVES ARE')
      DO 600 I=1,40
      WRITE (6,8100) I,PROB(I),MOVES(I)
 8100 FORMAT (' ',I3,5X,F10.8,10X,I10)
  600 CONTINUE
      IPATH=0
      PSUM=0.0
      DO 700 JJ=1,40
      PSUM=PSUM+PROB(JJ)
      IPATH=IPATH+MOVES(JJ)
  700 CONTINUE
C
C     THIS OUTPUT ENABLES A VISUAL CHECK TO DETERMINE IF
C     THE MASS FUNCTION SUMS TO ONE AND HOW MANY BRANCHES
C     EXISTED IN THE ENTIRE TREE.
C
      WRITE (6,8105) PSUM,IPATH
 8105 FORMAT ('0','SUM OF PROBABILITIES IS',F10.8,'TOTAL PAT
     CHS COUNTED',I10)
      STOP
      END
```

```
      SUBROUTINE RELOAD(LNCHRS,MOVES)
      DIMENSION LNCHRS(9)
C
C     THIS SUBROUTINE RELOADS THE EMPTY LAUNCHERS AND COUNTS
C     THE NUMBER OF MOVES REQUIRED.  WHEN THE BARN IS EMPTY,
C     THE FINAL TOTAL OF MOVES IS OUTPUT.  THE RELOAD
C     DOCTRINE IS TO MOVE ONLY ENOUGH MISSILES TO REFILL
C     THE LAUNCHERS.
C
      LCHEPT=0
      LSTORE=0
      DO 5 MI=1,3
      IF (LNCHRS(MI).EQ.0) LCHEPT=LCHEPT+1
    5 CONTINUE
      DO 10 MJ=4,9
      IF (LNCHRS(MJ).NE.0) LSTORE=LSTORE+1
   10 CONTINUE
      IF (LCHEPT.EQ.0.OR.LSTORE.EQ.0) RETURN
      IF (LCHEPT.LE.LSTORE) GO TO 70
      GO TO (15,20,40),LCHEPT
   15 RETURN
   20 IF (LNCHRS(3).NE.0) GO TO 33
      DO 25 MK=4,9
```

```
      IF (LNCHRS(MK).NE.0) GO TO 30
   25 CONTINUE
   30 LNCHRS(3)=LNCHRS(MK)
      LNCHRS(MK)=0
      MOVES=MOVES +(MK-3)
      RETURN
   33 LNCHRS(2)=LNCHRS(3)
      DO 35 MKK=4,9
      IF (LNCHRS(MKK).NE.0) GO TO 37
   35 CONTINUE
   37 LNCHRS (3)=LNCHRS(MKK)
      LNCHRS(MKK)=0
      MOVES =MOVES+1+(MKK-3)
      RETURN
   40 IF (LSTCRE.NE.1) GO TO 55
   43 DO 45 ML=4,9
      IF (LNCHRS(ML).NE.0) GO TO 50
   45 CONTINUE
   50 LNCHRS(3)=LNCHRS(ML)
      LNCHRS(ML)=0
      MOVES=MOVES+(ML-3)
      RETURN
   55 DO 60 MM=4,9
      IF (LNCHRS(MM).NE.0) GO TO 65
   60 CONTINUE
   65 LNCHRS(2)=LNCHRS(MM)
      LNCHRS(MM)=0
      MOVES=MOVES+(MM-2)
      GO TO 43
   70 DO 100 MN=1,3
      IF (LNCHRS(MN).NE.0) GO TO 100
      MIN=MN+1
      DO 75 MII=MIN,9
      IF (LNCHRS(MII).NE.0) GO TO 80
   75 CONTINUE
   80 LNCHRS(MN)=LNCHRS(MII)
      LNCHRS(MII)=0
      MOVES=MOVES+(MII-MN)
  100 CONTINUE
      RETURN
      END
```

COMPUTER PROGRAM II

```
      DIMENSION MISSIL(9),PROPOR(5),INARAY(9),KTGTCT(5),
     CSLIMIT(4),PWRHDS(5),PMOVE(40),JMOVES(40)
      DATA KSEED/191/,KTGTCT/5*0/,JMOVES/40*0/,WRHDS/0.0/,
     CLOCKON/0/,RMOVES/0.0/
C
C     READ IN THE NUMBER OF MISSILES IN THE SECTION.
C
      READ (5,1050) IBASLD
 1050 FORMAT (' ',I1)
C
C     READ IN THE INITIAL ORDER OF THE WARHEADS.
C
      READ (5,1200) INARAY
 1200 FORMAT (9I2)
C
C     READ IN THE PROBABILITIES OF A REQUEST FOR A
C     PARTICULAR WARHEAD.  THE FIRST PROBABILITY IS FOR THE
C     TYPE 1 WARHEAD, THE SECOND PROBABILITY IS FOR THE TYPE
C     2 WARHEAD,ETC.  FIVE NUMBERS MUST BE INPUT.  THE FIVE
C     NUMBERS MUST ADD TO ONE.
C
      READ (5,1300) PROPOR
 1300 FORMAT (5F10.3)
C
C     THIS OUTPUT ENSURES THAT THE INITIAL ARRAY WAS INPUT
C     CORRECTLY.
C
      WRITE (6,1400) INARAY
 1400 FORMAT (' ','THE INITIAL ARRAY IS',9I2)
      SLIMIT(1)=PROPOR(1)
      DO 10 J=2,4
      SLIMIT(J)=SLIMIT(J-1)+PROPOR(J)
   10 CONTINUE
C
C     THIS OUTPUT ENSURES THAT THE WARHEAD PROBABILITIES
C     WERE INPUT PROPERLY.
C
      WRITE (6,1600) (PROPOR(NNN),NNN=1,5)
 1600 FORMAT (' ','THE WARHEAD PROBABILITIES ARE',5F10.3)
C
C     THIS SIMULATION WILL BE RUN 10,000 TIMES.  EACH RUN
C     WILL BE WITH A DIFFERENT RANDOM STRING OF REQUESTS
C     FOR WARHEADS.
C
      DO 50 K=1,10000
      DO 14 KK=1,9
      MISSIL(KK)=INARAY(KK)
   14 CONTINUE
      NCRIT=0
      NCOUNT=0
   15 CALL TARGET(SLIMIT,LOCKON,KTGTCT,KSEED)
      DO 20 N=1,3
C
C     THIS CHECKS IF A LAUNCHER CONTAINS A WARHEAD OF THE
C     TYPE REQUESTED.  THE LAUNCHERS ARE CHECKED STARTING
C     WITH THE ONE CLOSEST TO THE STORAGE BARNS, THEN
C     CHECKING THE NEXT LAUNCHER OVER, AND FINALLY
C     CHECKING THE LAUNCHER FARTHEST FROM THE BARN.
C
C
      IF (MISSIL(4-N).EQ.LOCKON) GO TO 30
   20 CONTINUE
      CALL RELOAD (MISSIL,NCRIT)
      GO TO 15
   30 MISSIL(4-N)=0
```

```
      NCOUNT=NCOUNT+1
      IF (NCOUNT.LT.IBASLD) GO TO 15
      JMOVES(NCRIT)=JMOVES(NCRIT)+1
   50 CONTINUE
      DO 60 II=1,5
      WRHDS=WRHDS+KTGTCT(II)
   60 CONTINUE
C
C     THIS OUTPUT ENABLES A VISUAL CHECK OF WHETHER THE
C     RANDOM STRING OF REQUESTS CONTAINS THE PROPER
C     PROPORTION OF TYPES.
C
      WRITE (6,1800)
 1800 FORMAT (' ','WARHEAD DISTRIBUTION IS')
      DO 70 IJ=1,5
      PWRHDS(IJ)=KTGTCT(IJ)/WRHDS
      WRITE (6,1900) IJ,PWRHDS(IJ)
 1900 FORMAT (' ',I10,5X,F10.8)
   70 CONTINUE
      DO 75 IL=1,40
      RMOVES=RMOVES+JMOVES(IL)
   75 CONTINUE
C
C     THIS OUTPUT PROVIDES THE MASS FUNCTION OF THE NUMBER
C     OF MOVES REQUIRED TO EMPTY THE STORAGE BARNS DURING
      THE 10,000 TRIALS.
C
      WRITE (6,2000)
 2000 FORMAT (' ','DISTRIBUTION OF MOVES IS')
      DO 80 IK=1,40
      PMOVE(IK)=JMOVES(IK)/RMOVES
      WRITE (6,1900) IK,PMOVE(IK)
   80 CONTINUE
      STOP
      END




      SUBROUTINE TARGET(POINTS,MEXP,JCOUNT,IX)
      DIMENSION POINTS(4),JCOUNT(5)
C
C     THIS SUBROUTINE GENERATES THE RANDOM STRING OF
C     WARHEAD REQUESTS.  A RANDOM NUMBER BETWEEN ZERO AND
C     ONE IS GENERATED AND THEN THIS NUMBER IS COMPARED
C     WITH THE INPUT WARHEAD PROBABILITIES IN ORDER TO
C     DETERMINE THE TYPE OF REQUEST GENERATED.
C
      IX=IX*65539
      YFL=0.5+IX*2.328306E-10
      DO 5 M=1,4
      IF (YFL.LE.POINTS(M)) GO TO 10
    5 CONTINUE
      MEXP=5
      GO TO 20
   10 MEXP=M
   20 JCOUNT(MEXP)=JCOUNT(MEXP)+1
      RETURN
      END




      SUBROUTINE RELOAD(LNCHRS,MOVES)
      DIMENSION LNCHRS(9)
C
C     THIS SUBROUTINE RELOADS THE EMPTY LAUNCHERS AND COUNTS
C     THE NUMBER OF MOVES REQUIRED.  WHEN THE BARN IS EMPTY,
```

```
C       THE FINAL TOTAL OF MOVES IS OUTPUT.  THE RELOAD
C       DOCTRINE IS TO MOVE ONLY ENOUGH MISSILES TO REFILL
C       THE LAUNCHERS.
C
        LCHEPT=0
        LSTORE=0
        DO 5 MI=1,3
        IF (LNCHRS(MI).EQ.0) LCHEPT=LCHEPT+1
      5 CONTINUE
        DO 10 MJ=4,9
        IF (LNCHRS(MJ).NE.0) LSTORE=LSTORE+1
     10 CONTINUE
        IF (LCHEPT.EQ.0.OR.LSTORE.EQ.0) RETURN
        IF (LCHEPT.LE.LSTORE) GO TO 70
        GO TO (15,20,40),LCHEPT
     15 RETURN
     20 IF (LNCHRS(3).NE.0) GO TO 33
        DO 25 MK=4,9
        IF (LNCHRS(MK).NE.0) GO TO 30
     25 CONTINUE
     30 LNCHRS(3)=LNCHRS(MK)
        LNCHRS(MK)=0
        MOVES=MOVES +(MK-3)
        RETURN
     33 LNCHRS(2)=LNCHRS(3)
        DO 35 MKK=4,9
        IF (LNCHRS(MKK).NE.0) GO TO 37
     35 CONTINUE
     37 LNCHRS (3)=LNCHRS(MKK)
        LNCHRS(MKK)=0
        MOVES =MOVES+1+(MKK-3)
        RETURN
     40 IF (LSTORE.NE.1) GO TO 55
     43 DO 45 ML=4,9
        IF (LNCHRS(ML).NE.0) GO TO 50
     45 CONTINUE
     50 LNCHRS(3)=LNCHRS(ML)
        LNCHRS(ML)=0
        MOVES=MOVES+(ML-3)
        RETURN
     55 DO 60 MM=4,9
        IF (LNCHRS(MM).NE.0) GO TO 65
     60 CONTINUE
     65 LNCHRS(2)=LNCHRS(MM)
        LNCHRS(MM)=0
        MOVES=MOVES+(MM-2)
        GO TO 43
     70 DO 100 MN=1,3
        IF (LNCHRS(MN).NE.0) GO TO 100
        MIN=MN+1
        DO 75 MII=MIN,9
        IF (LNCHRS(MII).NE.0) GO TO 80
     75 CONTINUE
     80 LNCHRS(MN)=LNCHRS(MII)
        LNCHRS(MII)=0
        MOVES=MOVES+(MII-MN)
    100 CONTINUE
        RETURN
        END
```

BIBLIOGRAPHY

1. Everett, H. III, Defense Models XI-Exhaustion of Interceptor Supplies, Arlington, Va., Lamda Corporation, 1968, p. 1.

2. Gregg, Wendell H., Major, U.S. Army, 490-38-0796, S-2 (Intelligence Officer), 13th Air Defense Artillery Group, interviewed by Michael L. Mulhern, Presidio of San Francisco, 140900 December 1973.

3. Headquarters, Department of the Army, Staff Officers' Field Manual — Nuclear Weapons Employment Effects Data, FM 101-31-3, February 1968, p. 16.

4. Hickman, Edgar P., and Hilton, James G., Probability and Statistical Analysis, Scranton, Intext Educational Publishers, 1971, p. 37.

5. Mahoney, William D., Chief Warrant Officer, U.S. Army, 568-46-1186, Air Defense Assembly Technician, 13th Air Defense Artillery Group, interviewed by Michael L. Mulhern, Presidio of San Francisco, 141000 December 1973.

6. Shere, Kenneth D., and Cohen, Edgar A., "A Defense Allocation Problem with Development Costs," Naval Research Logistics Quarterly, v. 19, pp. 525-537, March 1973.

7. Siegel, Sidney, Nonparametric Statistics for the Behavioral Sciences, New York, McGraw-Hill Book Company, 1956, p. 48.

8. Soland, Richard M., "Optimal Defensive Missile Allocation: A Discrete Min-Max Problem," Operations Research, v. 21, pp. 590-596, March-April 1973.

9. United States Army Air Defense School, Surface-to-Air Missile Analyzing Devices, Student Handout No. CS3. 76209, October 1971, p. 3.

10. Waddell, M. C., Air Battle Analyzer Handbook, Silver Springs, Maryland, Applied Physics Laboratory, 1963, pp. 25-27.

11. Zehna, Peter W., Probability Distributions and Statistics, Boston, Allyn and Bacon, Inc., 1970, pp. 424-430.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	2
4.	CPT Michael L. Mulhern, USA 1102 North Trinity Decatur, Texas 76234	1
5.	Chief of Personnel Operations ATTN: OPXC Department of the Army Washington, D.C. 20315	1
6.	Assoc. Prof. James G. Taylor, Code 55TW Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
7.	Chief of Naval Personnel (Pers 11b) Department of the Navy Washington, D.C. 20370	1

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Ordering in a Sequential Storage System		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1974
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Michael Lynn Mulhern		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1974
		13. NUMBER OF PAGES 83
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Sequential Storage
Nike-Hercules Missile
Warhead Allocation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A missile system in which L launchers were available in a row and S spare missiles were stored in a continuation of this row was considered. Additionally, an assortment of warheads was available, but only one could be mounted in each missile. The ability of this system to meet a demand for a particular warhead was dependent on the mix of warheads in the L+S missiles and the order in which the missiles with their varying warheads

DD FORM 1473 1 JAN 73 (Page 1) EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(20. ABSTRACT continued)

were mounted on the launchers and in storage. Since the missiles were in a continuous row, some missiles would not be available for firing before the missiles in front of them had been fired.

Because of the myriad possible permutations of the L+S warheads, determination of optimal ordering of the warheads by simulation was found to be too time-consuming. Hence, analytic expressions were developed after an appropriate measure of effectiveness was devised. A computer program was developed to perform the recursive calculations necessary.

Thesis
M8856 Mulhern
c.1
149187
Ordering in a sequential storage system.

Thesis
M8856 Mulhern
c.1
149187
Ordering in a sequential storage system.

thesM8856
Ordering in a sequential storage system.
3 2768 000 99335 6
DUDLEY KNOX LIBRARY